Available online at www.sciencedirect.com
SCIENCE @ DIRECT®
ELSEVIER
Linear Algebra and its Applications 375 (2003) 221–243
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

Semidefinite programming and matrix scaling over the semidefinite cone

Bahman Kalantari

Department of Computer Science, Hill Center for Mathematical Sciences, Busch Campus, Rutgers University, New Brunswick, NJ 08903, USA

Received 31 October 2001; accepted 3 June 2003

Submitted by H. Schneider

Abstract

Let E be the Hilbert space of real symmetric matrices with block diagonal form $\mathrm{diag}(A, M)$, where A is $n \times n$, and M is an $l \times l$ diagonal matrix, with the inner product $\langle x, y\rangle \equiv \mathrm{Trace}(xy)$. We assume $n + l \geqslant 1$, i.e. allow $n = 0$ or $l = 0$. Given $x \in E$, we write $x \succeq 0$ ($x \succ 0$) if it is positive semidefinite (positive definite). Let $Q : E \to E$ be a symmetric positive semidefinite linear operator, and $\mu = \min\{\phi(x) = 0.5\,\mathrm{Trace}(xQx) : \|x\| = 1, x \succeq 0\}$. The problem of testing if $\mu = 0$ is a significant problem called *Homogeneous Programming*. On the one hand the feasibility problem in semidefinite programming (SDP) can be formulated as a Homogeneous Programming problem. On the other hand it is related to the generalization of the classic problem of *Matrix Scaling*. Let $\epsilon \in (0, 1)$ be a given accuracy, $u = Qe - e$, e the identity matrix in E, and $N = n + l$. We describe a path-following algorithm that in case $\mu = 0$, in $\mathrm{O}(\sqrt{N} \ln[N\|u\|/\epsilon])$ Newton iterations produces $d \succeq 0$, $\|d\| = 1$ such that $\phi(d) \leqslant \epsilon$. If $\mu > 0$, in $\mathrm{O}(\sqrt{N} \ln[N\|u\|/\mu] + \ln\ln(1/\epsilon))$ Newton iterations the algorithm produces $d \succ 0$ such that $\|DQDe - e\| \leqslant \epsilon$, where D is the operator that maps $w \in E$ to $d^{1/2}wd^{1/2}$. Moreover, we use the algorithm to prove: $\mu > 0$, if and only if there exists $d \succ 0$ such that $DQDe = e$, if and only if there exists $d \succ 0$ such that $Qd \succ 0$. Thus via this duality the Matrix Scaling problem is a natural *dual* to the feasibility problem in SDP. This duality also implies that in Blum et al. [Bull. AMS 21 (1989) 1] real number model of computation the decision problem of testing the solvability of Matrix Scaling is both in NP and Co-NP. Although the above complexities can be deduced from our path-following algorithm for general self-concordant Homogeneous Programming and for Matrix Scaling obtained in [Scaling dualities and self-concordant homogeneous programming in finite dimensional spaces, Technical Report LCSR-TR-359, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1999], for the problems considered here the present analysis is quite

E-mail address: kalantar@cs.rutgers.edu

doi:10.1016/S0024-3795(03)00664-5

elementary, short, and complete. This simplicity is mainly due to a new inequality derived in this paper that relates the norm of *scaled* quantities at two successive Newton iterations and implies a quadratic rate of convergence. The present algorithm is not only a simple path-following algorithm for testing the solvability of feasibility problem in SDP, but is also capable of testing solvability of the Matrix Scaling problem. When $n = 0$, the algorithm reduces to the diagonal Matrix Scaling/Linear Programming algorithm of Khachiyan and Kalantari [SIAM J. Optim. 4 (1992) 668]. As in the case of LP the algorithm of this paper can be used to solve the general SDP problem.

AMS classification: 90C25; 68Q25; 65F35; 15A48

Keywords: Semidefinite programming; Matrix scaling; Newton's method; Interior-point method; Positive semidefinite linear operators

1. Introduction

It is known [7] that given an $n \times n$ real symmetric positive semidefinite matrix A, either there exists a nonnegative nonzero vector $d \in \Re^n$ such that $d^{\mathrm{T}}Ad = 0$, or there exists a positive vector $d \in \Re^n$ such that DAD is quasi-doubly stochastic, i.e. its row and columns sums are all ones, where $D = \mathrm{diag}(d) = \mathrm{diag}(d_1, \ldots, d_n)$. This is called the *Scaling Equation* or more precisely *Diagonal Scaling Equation*. We shall refer to the problem of testing the solvability of the first as *Homogeneous Programming*, and the second as *Matrix Scaling*. Clearly, $d^{\mathrm{T}}Ad = 0$ if and only if $Ad = 0$ which implies that the two problems are not simultaneously solvable. The ϵ-approximate solvability of both problems can be established in a polynomial number of iterations of a path-following algorithm whose complexity is polynomial in n, $\ln(1/\epsilon)$, and the size of encoding of constants that depend on the input matrix A [11,13]. For obtaining an ϵ-approximate solution of Matrix Scaling the algorithm requires a lower bound on the quantity μ defined as the minimum of the function $\frac{1}{2}x^{\mathrm{T}}Ax$ over the intersection of the unit sphere and the nonnegativity cone.

It is also known that the feasibility problem in linear programming, in the absence of a recession direction, can be converted to that of testing if $Ax = 0$ has a nonnegative nontrivial solution for some symmetric positive semidefinite matrix A. More generally, any linear programming problem with rational inputs can be converted to this canonical Homogeneous Programming problem. Thus, at least in theory, the Matrix Scaling problem is a natural dual problem to linear programming and as rich of a problem as linear programming itself. Moreover, in [7] it is proved that the Homogeneous Programming problem, i.e. testing if $d^{\mathrm{T}}Ad = 0$ when A is only symmetric, but not positive semidefinite, is NP-complete. Thus Homogeneous Programming is a fundamental problem in theoretical computer science as well.

The goal of the present paper is to extend the definition of Homogeneous Programming and Matrix Scaling, as well as the path-following algorithm of [13] to the case where the underlying space is the Hilbert space of real symmetric matrices, the underlying cone the cone of positive semidefinite symmetric matrices, and where the matrix A above is replaced with a symmetric positive semidefinite linear operator.

In Section 2 we first formally define Homogeneous Programming and Matrix Scaling over the semidefinite cone as well as their ϵ-approximate versions. In Section 3 we give the ingredients and an overview of our path-following algorithm and a comparison to the existing interior-point path-following algorithms. In Section 4 we define Newton direction, iterate, decrement, and state relationships to their scaled versions. In Section 5 we prove the main results needed to state the path-following algorithm. In Section 6 we formally define the path-following algorithm and use the main results to obtain a complexity bound for solving the desired ϵ-approximate problems. In Section 7 we use the algorithm to prove the solvability of the corresponding *Scaling Equation*. We also prove theorems of the alternative, called *scaling dualities*. In Section 8 we show how to reduce the problem of obtaining a lower bound, analogous to the quantity μ defined above, to a semidefinite programming problem. In Section 9 we consider conversion of the feasibility problem in semidefinite programming (SDP) to that of a corresponding Homogeneous Programming problem. We close with concluding remarks.

2. Homogeneous programming and matrix scaling over the semidefinite cone

Consider S^n, the set of $n \times n$ real symmetric matrices, and let $\mathrm{Tr}(\cdot)$ denote the trace function. The notation $x \succeq 0$ as usual means that x lies in S^n_+, the set of positive semidefinite matrices in S^n. Let

$$E = \left\{ x \in S^{n+l} : x = \begin{bmatrix} A & 0 \\ 0 & M \end{bmatrix},\ A \in S^n,\ M = \mathrm{diag}(u),\ u \in \Re^l \right\} \tag{2.1}$$

be the Hilbert space where the inner product of $x, y \in E$ is defined as $\langle x, y \rangle = \mathrm{Tr}(xy)$. The corresponding induced norm is $\|x\| = \sqrt{\mathrm{Tr}(x^2)}$. Let $N = n + l$. We assume $N \geqslant 1$, i.e. allow $n = 0$ or $l = 0$.

Let Q be a given symmetric positive semidefinite linear operator in $L(E, E)$, the set of linear transformations from E into itself. Thus, $Q = Q^{\mathrm{T}}$, and for all $x \in E$, $\mathrm{Tr}(xQx) \geqslant 0$. Let

$$\phi(x) = \frac{1}{2}\mathrm{Tr}(xQx). \tag{2.2}$$

We define *Homogeneous Programming* to be the problem of testing if $\phi(x)$ has a nontrivial zero over the nonnegativity cone $K = \{x : x \succeq 0\}$. Since Q is symmetric positive semidefinite, $\phi(x) = 0$ if and only $Qx = 0$. This immediately implies that Homogeneous Programming is solvable if and only if the following SDP is feasible

$$\{x : Qx = 0, \mathrm{Tr}(x) = 1, x \succeq 0\}. \tag{2.3}$$

However, we are interested in the formulation of Homogeneous Programming as the problem of testing if $\mu = 0$, where

$$\mu = \min\left\{\phi(x) = \tfrac{1}{2}\mathrm{Trace}(xQx) : \|x\| = 1, x \succeq 0\right\}. \tag{2.4}$$

Computationally, we are interested in approximate solutions defined below.

Definition 2.1. For a given $\epsilon \in (0, 1)$, we define ϵ-approximate Homogeneous Programming to be the problem of computing $d \in E$ such that $d \succeq 0$, $\|d\| = 1$, and $\phi(d) \leqslant \epsilon$; or proving its unsolvability.

Each $d \succ 0$ induces an operator D in $L(E, E)$ defined by the mapping

$$Dw = d^{1/2}wd^{1/2}. \tag{2.5}$$

The *Matrix Scaling* problem is to test if there exists $d \succ 0$ such that the *Scaling Equation*

$$DQDe = e, \tag{2.6}$$

holds, where e is the identity matrix in E. An interpretation of the Scaling Equation is that for some $d \succ 0$ the linear operator DQD has e as its eigenvector with eigenvalue of 1 (equivalently e is a fixed-point of the map DQD). It is easy to see that the Scaling Equation is equivalent to the following

$$Qd = d^{-1}. \tag{2.7}$$

As in the case of Homogeneous Programming, computationally we are interested in approximate solutions defined below.

Definition 2.2. For a given $\epsilon \in (0, 1)$, we define ϵ-approximate Matrix Scaling to be the problem of computing $d \succ 0$ such that

$$\|DQDe - e\| \leqslant \epsilon, \tag{2.8}$$

or proving its unsolvability.

We shall describe a simple path-following algorithm that given $\epsilon \in (0, 1)$ either solves the ϵ-approximate Homogeneous Programming, or ϵ-approximate Matrix Scaling problem. Moreover, we show that ϕ has no nontrivial nonnegative zero, if and only if there exists $d \succ 0$ such that $DQDe = e$, if and only if there exists $d \succ 0$ such that $Qd \succ 0$. These result can be interpreted as two theorems of the alternative, called *scaling dualities*:

$$\text{(scaling duality)} \quad \begin{array}{l} \exists d \succeq 0, \ \ d \neq 0, \quad Qd = 0; \\ \text{or } \exists d \succ 0, \ \ DQDe = e; \ \ \text{not both.} \end{array} \tag{2.9}$$

$$\text{(Gordan's duality)} \quad \begin{array}{l} \exists d \succeq 0, \ \ d \neq 0, \quad Qd = 0; \\ \text{or } \exists d \succ 0, \ \ DQDe \succ 0; \ \ \text{not both.} \end{array} \tag{2.10}$$

Both dualities imply analogous dualities for positive semidefinite symmetric matrices, where the Scaling Equation reduces the diagonal scaling equation. More specifically, the dualities hold when Q is replaced with any positive semidefinite symmetric matrix A, e by the vector of ones, and the semidefinite cone with the nonnegativity cone in the Euclidean space.

The diagonal scaling for positive definite symmetric matrices was first proved by Marshall and Olkin [14]. Inspired by the work of Karmarkar [12], the diagonal Matrix Scaling Duality was proved independently in Kalantari [7], and in more generality. More generally, in the context of homogeneous functions the diagonal Matrix Scaling Duality as well as Gordan's duality were proved in Kalantari [8,9]. The dualities proved in this paper are in fact very special cases of far more general dualities, called *scaling dualities*, proved in Kalantari [10]. The polynomial-time solvability for positive semidefinite Matrix Scaling was first established by Khachiyan and Kalantari [13]. O'Leary [17] considers matrix scaling of symmetric positive definite matrices over other orthants and second-order cone programming formulation of the problem. The path-following algorithm of this paper is essentially a special case of a path-following algorithm for general self-concordant Homogeneous Programming and for Matrix Scaling problem, derived in Kalantari [10]. The analysis of the general algorithm is complicated and relies on many results, in particular the self-concordance theory of Nesterov and Nemirovskii. While for the problems considered in this paper the same complexities can be stated from the path-following algorithm of Kalantari [10], the present analysis is quite elementary and short, while complete. This simplicity is mainly due to a new inequality that relates *scaled* quantities at two successive Newton iterations (Lemma 5.3) thereby implying quadratic rate of convergence while solving Homogeneous Programming or Matrix Scaling. When $n = 0$, the present algorithm reduces to the Matrix Scaling/linear programming algorithm of Khachiyan and Kalantari [13]. The generalization here however is novel and in the spirit of a rederivation of that algorithm described in [10] (Section 11.3).

In view of the fact that the feasibility problem in SDP can be converted into a homogeneous feasibility problem (Section 9), the path-following algorithm of this paper is not only capable of solving this feasibility problem, but a problem that has not been considered in the the existing semidefinite programming literature, namely Matrix Scaling over the semidefinite cone. As in the case of linear programming, the scaling problem happens to be a natural *dual* to SDP. For the relationship between general convex programming and the general notion of scaling, see [10].

3. Ingredients and overview of a path-following algorithm

Consider the Hilbert space E defined in (2.1), and a given symmetric positive semidefinite linear operator Q in $L(E, E)$. Let

$$F(x) = -\ln\det(x), \quad \psi(x) = \phi(x) + F(x). \tag{3.1}$$

Given $d \succ 0$, let D be the operator in $L(E, E)$ defined in (2.5). Formally, $D = \nabla^2 F(d)^{-1/2}$ (see Proposition 3.16 in [10]), where ∇^2 denotes the Hessian. But this fact could be ignored altogether in the analysis of the algorithm to be presented.

If the equation $DQDe = e$ is solvable for some $d \succ 0$, then since $De = d$, we get

$$\nabla\psi(d) = Qd - d^{-1} = 0, \tag{3.2}$$

where ∇ denotes the gradient. Conversely, if the gradient is zero at a positive d then the Scaling Equation is solvable. Since $\psi(x)$ is strictly convex, the above also implies the uniqueness of the solution to the Scaling Equation (2.6), if it exists.

Thus the (ϵ-approximate) Matrix Scaling problem is the problem of computing an (ϵ-approximate) stationary point of ψ, but in a *scaled* setting. It is this scaled format which makes it possible to state our path-following algorithm.

Let $u = e - Qe$, and for each $t \in (0, \infty)$, define

$$f^t(x) = t\phi(x) + t\,\mathrm{Tr}(ux) + F(x). \tag{3.3}$$

Note that when $t = 1$, $\nabla f^t(e) = Qe - e - Qe + e = 0$. Thus e is the minimizer of $f^1(x)$ over the semidefinite cone. The path of approximate minimizers of the family f^t, the so-called central path, as t approaches zero, will be used to decide which of the two ϵ-approximate problems is solvable. This follows from the properties of the underlying problem as opposed to the application of known interior-point results. Algorithmically, and unlike the existing path-following algorithms, instead of using an approximation to the central path, we make use of approximate scaled stationary points of f^t. To give a more precise description of our path-following algorithm and a more clear distinction between our path-following and the existing ones we first need the following:

Definition 3.1. Given $d \succ 0$, consider its corresponding operator D, see (2.5), and define the induced (or scaled) functions to be

$$\phi_d(x) = \phi(Dx), \quad \psi_d(x) = \psi(Dx), \quad f_d^t(x) = f^t(Dx). \tag{3.4}$$

Thus for each $d \succ 0$ we have three new functions. From the chain rule the induced (or scaled) gradients are

$$\begin{aligned} \nabla\phi_d(x) &= D\nabla\phi(Dx), \\ \nabla\psi_d(x) &= D\nabla\psi(Dx), \\ \nabla f_d^t(x) &= D\nabla f^t(Dx). \end{aligned} \tag{3.5}$$

We now describe the properties of the parameterized family, f^t, $t \in (0, \infty)$. These properties are deducible from the results in Section 5 which proves the main results in this paper. It turns out that for each $t \in (0, \infty)$ the minimizer of f^t:

$$d_t^* = \mathrm{argmin}\{f^t(x) : x \succ 0\} \tag{3.6}$$

is well defined. Thus the central-path, $\{d_t^* : t \in (0, \infty)\}$, exists. Moreover, if $\mu = 0$, it will follow that $\phi(d_t^*/\|d_t^*\|) = \mathrm{O}(t)$. This implies that the projection of the central-path on the unit sphere, i.e.

$$\left\{\frac{d_t^*}{\|d_t^*\|} : t \in (0, \infty)\right\}, \tag{3.7}$$

as t approaches zero, will converge to a nonnegative zero of $\phi(x)$ over the unit sphere. The limit may of course turn out to be positive semidefinite. If $\mu > 0$, then as t approaches 0, $\hat{d}_t^* = \sqrt{t}d_t^*$ converges to d^* satisfying $D^*QD^*e = e$, i.e. the unique solution to the Matrix Scaling problem. More importantly, through the path-following algorithm, it is shown that instead of computing d_t^*, or trying to stay in a standard neighborhood of it, we can compute any approximate minimizer, d, of f^t so that $\|\nabla f_d^t(e)\|$ is sufficiently small, e.g. any number less than 1/2.

The path-following algorithm consists of two phases. However, if we are interested in solving ϵ-approximate Homogeneous Programming, we only need the first phase.

In Phase I, given $\epsilon \in (0, 1)$, by applying Lemma 5.1 and 5.2 we first determine an appropriate $t_* \in (0, 1)$ such that if $\mu = 0$, then computing a point d with $\|\nabla f_d^{t*}(e)\| < 1/2$ will guarantee that $\phi(d/\|d\|) \leqslant \epsilon$.

To compute the desired d above, we make use of key result in Lemma 5.3 and its consequence Lemma 5.5. Lemma 5.3 states that if for a given t, we have available a point $d \succ 0$ satisfying the inequality $\|\nabla f_d^t(e)\| < 1$, then the application of Newton's method as applied to $f^t(x)$ will result in the Newton iterate d', necessarily positive, satisfying $\|\nabla f_{d'}^t(e)\| \leqslant \|\nabla f_d^t(e)\|^2$ (i.e. the scaled gradients at e decrease at a quadratic rate). This result which also applies to $\psi(x)$ (Corollary 5.4), implies Lemma 5.5 showing that once we have $t \in (0, \infty)$ and $d \succ 0$ such that $\|\nabla f_d^t(e)\| < \gamma_0$, a given number in (0, 0.5), then setting $t' = tr_*$, with $r_* = (\sqrt{N} - \gamma_0/\sqrt{N} - \gamma_0^2)$ will imply that the Newton iterate d' above will also satisfy $\|\nabla f_{d'}^{t'}(e)\| < \gamma_0$. Hence the process of reducing t can be repeated. Since $t = 1$, and $u = e - Qe$, then e is a stationary point of f^t, and thus $\|\nabla f_e^1(e)\| = 0$, the process of reduction of t to reach the desired t_* may begin at $t = 1$.

If the first phase does not end up with an ϵ-approximate solution of the Homogeneous Programming problem, then it must have resulted in a point d such that

$$\|tDQDe - e\| < \gamma_0. \tag{3.8}$$

The above is equivalent to

$$\|\widehat{D}Q\widehat{D}e - e\| < \gamma_0, \quad \hat{d} = \sqrt{t}d. \tag{3.9}$$

Thus in this case we end up with an γ_0-approximate Matrix Scaling solution. This condition already guarantees the solvability of the exact Matrix Scaling problem (Theorem 7.1). Phase II does not require a path-following scheme. To obtain a desired ϵ-approximate solution requires repeated application of Newton's method to the function $\psi(x)$, starting with $\hat{d}$ described above. Quadratic convergence is guaranteed by Corollary 5.4 of Lemma 5.3. If however, our goal is to solve the ϵ-approximate Matrix Scaling problem, then we need an a priori lower bound on μ, if it is positive. While the computation of this lower bound is not easy, we prove that it is equivalent to the problem of obtaining a lower bound on the objective value of a semidefinite programming problem. However, when Q is known to be positive definite, as opposed to positive semidefinite, then any lower bound on the minimum eigenvalue of Q can be used. In practice, when solving the ϵ-approximate Matrix Scaling without an a priori estimate on μ, it suffices to repeatedly solve the ϵ-approximate Homogeneous Programming via Phase I, while ϵ is halved after each successful attempt. We explain this in more detail in Remark 2 of Section 6.

Here we would like to mention some facts about our path-following algorithm. These will become more clear to the reader as he/she will gain a deeper understanding of the algorithm. Although the path-following algorithm considered in this paper has similarities to the class of so-called barrier-generated path-following algorithms, it is fundamentally different than those algorithms, and other existing path-following algorithms.

Firstly, and ironically, if $\phi(x)$ is replaced with a linear functional, the path-following algorithm will fail to exhibit the above mentioned properties. The reader having carefully examined the algorithm will notice that it is important for homogeneous degree of $\phi(x)$ to be greater than one (as is two for the case of quadratic) and replacing $\phi(x)$ by a linear function will make the algorithm fail.

Secondly, unlike the typical barrier-generated path-following algorithms the domain of the optimization of f^t is the unbounded cone of nonnegativity. In Nesterov and Nemirovskii's book they consider barrier methods over bounded domains. When the domain is bounded and the barrier makes the function value approach infinity on the boundary, then the attainment of the minimizer can trivially be argued. However, in the case of unbounded domain such as the minimization of f^t, it not even obvious why the corresponding central-path, $\{d_t^* : t \in (0, \infty)\}$, should exist. For instance, consider the case of minimization of $f^t(x)$ for any positive value of t different than one. The infimum is attained but it takes an argument to prove it. The main results of this paper in particular implies the existence of the minimum and hence the central-path.

Thirdly, unlike the existing barrier-generated path-following methods, in our path-following method the central-path by itself is of no direct significance. Rather, its projection onto the unit sphere, i.e. $\{d_t^*/\|d_t^*\| : t \in (0, 1]\}$ as t approaches zero, will either converge to a nonnegative zero of $\phi(x)$ over the unit sphere, or to a positive point $\bar{d}$ such that $\bar{D}Q\bar{D}e$ is a scalar multiple of e. Such $\bar{d}$ can easily be scaled to give a solution d to the Scaling Equation $DQDe = e$. Equivalently, the solution of the Scaling Equation is also the limit of $\hat{d}_t^* = \sqrt{t}d_t^*$ as t approaches 0.

Fourthly, issues regarding the approximation of the projected central-path, and the significance of this approximation, as well as their application in terms of ϵ-approximate version of the two problems are issues whose answers rely on the scaling dualities, bounds, and sensitivity analysis, specifically developed in this paper, while using elementary linear algebra on symmetric matrices, as opposed to the mere application of general results from semidefinite programming, convex programming, or those implied by the theory of self-concordance.

4. Newton direction, iterate, decrement, and their scaled versions

Here we consider Newton direction, Newton iterate, and Newton decrement for minimization of the function $f^t(x)$ at a given positive point d, as well as those quantities for minimization of the function $f_d^t(x)$ at e.

Definition 4.1. Given $t \in (0, \infty)$, the Newton direction for minimization of f^t at a given $d \succ 0$, denoted by $y_t(d)$, is the solution to

$$\begin{aligned} &\nabla^2 f^t(d) y_t(d) = -\nabla f^t(d), \quad \text{i.e.} \\ &(tQ + D^{-2}) y_t(d) = -(tQd + tu - d^{-1}). \end{aligned} \tag{4.1}$$

The Newton iterate and the Newton decrement at d are, respectively

$$d_t' = d + y_t(d), \quad \lambda_t(d) = \sqrt{\mathrm{Tr}(y_t(d)\nabla^2 f^t(d) y_t(d))}. \tag{4.2}$$

As usual if $P_2(x)$ is the quadratic approximation of $f^t(x)$ at a given point $d \succ 0$, then the unconstrained minimizer, d_t', of $P_2(x)$ is the Newton iterate at d. The Newton direction is $y_t(d) = d_t' - d$. The Newton decrement is $(2[f^t(d) - P_2(d_t')])^{0.5}$ (see [15] for more general definition of Newton decrement for self-concordant functions). All we need in this paper is the formal definition of the quantities given above.

We also have corresponding quantities with respect to $\psi(x)$.

Since D^{-1} is the operator that maps $w \in E$ into $d^{-1/2} w d^{-1/2}$, we have

$$\begin{aligned} \mathrm{Tr}(xD^{-2}x) = \mathrm{Tr}(xd^{-1}xd^{-1}) &= \mathrm{Tr}(d^{-1/2}xd^{-1}xd^{-1/2}) \\ &= \mathrm{Tr}((d^{-1/2}xd^{-1/2})^2), \end{aligned} \tag{4.3}$$

it follows that $Q + D^{-2}$ is positive definite. Hence $y_t(d)$ is well-defined.

Proposition 4.2 (Scaled Newton direction, iterate, and decrement). *Let $z_t = D^{-1} y_t(d)$. Then z_t is the solution to*

$$\begin{aligned} &\nabla^2 f_d^t(e) z_t = -\nabla f_d^t(e), \quad \text{i.e.} \\ &(tDQD + I) z_t = -(tDQDe + tDu - e). \end{aligned} \tag{4.4}$$

Thus the Newton direction and iterate with respect to f_d^t at e are z_t and $e + z_t$, respectively. The corresponding Newton decrement remains invariant.

Proof. Substituting $y_t(d) = Dz_t$ in (4.1), applying the operator D, and also since $d = De$, $D^{-1}d = e$, we get (4.4). It is also easy to show that $\lambda_t(d) = \sqrt{\mathrm{Tr}(z_t \nabla^2 f_d^t(e) z_t)}$. □

5. Main results

The reader should keep in mind that this section makes use of elementary results on symmetric positive (semi)definite matrices as applied to the operator Q and elements of E.

Lemma 5.1. *If* $\mu = 0$, *then for all* $d \succ 0$, $\|DQDe - e\| \geqslant 1$.

Proof. Suppose there exists $d \succ 0$ such that $\|DQDe - e\| < 1$. Let $y = DQDe$. We claim that $y \succ 0$. Let $w \neq 0$ be an arbitrary point in $\Re^N$ and $\|\cdot\|_2$ be the 2-norm.

$$\begin{aligned} w^{\mathrm{T}}w - w^{\mathrm{T}}yw = w^{\mathrm{T}}(e-y)w &\leqslant (w^{\mathrm{T}}w)\|y-e\|_2 \\ &\leqslant (w^{\mathrm{T}}w)\|y-e\| < w^{\mathrm{T}}w. \end{aligned} \tag{5.1}$$

Thus, $w^{\mathrm{T}}yw > 0$. Hence y is positive definite. Now since $y \succ 0$, $D^{-1}y \succ 0$. This implies $QDe \succ 0$. But since $De = d$, this implies that $Qd \succ 0$. We claim that the positivity of Qd implies that there does not exist $x \succeq 0$, $x \neq 0$ such that $Qx = 0$. Otherwise,

$$\mathrm{Tr}(xQd) = 0 = \mathrm{Tr}((Qd)^{1/2}x(Qd)^{1/2}). \tag{5.2}$$

Since the matrix $(Qd)^{1/2}x(Qd)^{1/2}$ is positive semidefinite and its trace is zero, it must be the zero matrix. But this implies $x = 0$, a contradiction. Thus, if $Qd \succ 0$, then $\mu > 0$. □

Lemma 5.2. *Let* γ *be a number in* $(0, 1]$. *Given* $t \in (0, 1]$, *suppose there exists* $d \succ 0$ *such that*

$$\|\nabla f_d^t(e)\| = \|tDQDe + tDu - e\| \leqslant \frac{1}{2}\gamma. \tag{5.3}$$

Let $\hat{d} = \sqrt{t}d$ *and* $\widehat{D} = \sqrt{t}D$. *If* $\|\widehat{D}Q\widehat{D}e - e\| \geqslant \gamma$, *then*

$$\phi\left(\frac{d}{\|d\|}\right) \leqslant C(\gamma)t, \tag{5.4}$$

where

$$C(\gamma) = \frac{1}{\gamma^2}\left[2N + \gamma\left(\sqrt{N}+1\right)\right]\|u\|^2. \tag{5.5}$$

Proof. Clearly,

$$\nabla f_d^t(e) = tDQDe + tDu - e = (\widehat{D}Q\widehat{D}e - e) + \sqrt{t}\widehat{D}u. \tag{5.6}$$

Taking inner product with e, using Cauchy–Schwarz inequality, and the bound in (5.3), we get

$$\langle e, \nabla f_d^t(e)\rangle = 2\phi(\hat{d}) + \sqrt{t}\langle u, \hat{d}\rangle - N \leqslant \|e\|\|\nabla f_d^t(e)\| \leqslant \frac{\sqrt{N}}{2}\gamma. \tag{5.7}$$

Since $\langle u, \hat{d}\rangle \leqslant \|u\|\|\hat{d}\|$, (5.7) implies,

$$2\phi(\hat{d}) \leqslant N + \frac{\sqrt{N}}{2}\gamma + \sqrt{t}\|\hat{d}\|\|u\|. \tag{5.8}$$

Next we show

$$\|D\| \leqslant \|d\|. \tag{5.9}$$

If $A \in S^n_+$, then $\|A\|^2 = \mathrm{Tr}(A^2) \leqslant \mathrm{Tr}(A)^2$. Using this, for any $x \in E$ we have,

$$\|Dx\| = \|d^{1/2}xd^{1/2}\| \leqslant \mathrm{Tr}(d^{1/2}xd^{1/2}) = \mathrm{Tr}(dx) = \langle d, x\rangle \leqslant \|d\|\|x\|. \tag{5.10}$$

Since given $a, b \in E$, $\|a - b\| \geqslant \|a\| - \|b\|$, from (5.3) and (5.6) we have,

$$\|\sqrt{t}\widehat{D}u\| \geqslant \|\widehat{D}Q\widehat{D}e - e\| - \|\nabla f^t_d(e)\| \geqslant \gamma - \frac{1}{2}\gamma = \frac{1}{2}\gamma. \tag{5.11}$$

Since $\|\hat{d}\|\|u\| \geqslant \|\widehat{D}\|\|u\| \geqslant \|\widehat{D}u\|$, from (5.11) we get,

$$\frac{1}{\|\hat{d}\|} \leqslant \frac{2\sqrt{t}\|u\|}{\gamma}. \tag{5.12}$$

Dividing the inequality in (5.8) by $\|\hat{d}\|^2$, using the bound in (5.12), and since $\phi(\hat{d})/\|\hat{d}\|^2 = \phi(d/\|d\|)$, we get the desired result. □

Lemma 5.3. *Assume $d \succ 0$, $t \in (0, \infty)$. Let $d'_t = d + y_t(d)$, $z_t = D^{-1}y_t(d)$. Then the following four statements hold*:

(i) $\mathrm{Tr}(\nabla f^t_{d'_t}(e)) = -\|z_t\|^2$.
(ii) *If* $\|z_t\| < 1$, *then* $d'_t \succ 0$.
(iii) *If* $\|z_t\| < 1$, *then* $-\nabla f^t_{d'_t}(e) \in S^n_+$.
(iv) *If* $\|\nabla f^t_d(e)\| < 1$, *and* $z'_t = D'^{-1}_t y_t(d'_t)$, *then we have*

$$\|z'_t\| \leqslant \lambda_t(d'_t) \leqslant \|\nabla f^t_{d'_t}(e)\| \leqslant \|z_t\|^2 \leqslant \lambda_t(d)^2 \leqslant \|\nabla f^t_d(e)\|^2. \tag{5.13}$$

Proof. (i) Consider the equation defining z_t (see (4.4)),

$$(tDQD + I)z_t = e - tDQDe - tDu. \tag{5.14}$$

Regrouping terms in (5.14) gives

$$tDQD(e + z_t) = e - z_t - tDu. \tag{5.15}$$

Recalling the definition of the operator D (see (2.5)), we have $D(e + z_t) = d + y_t(d) = d'_t = D'_t e$, where D' is the operator that corresponds to d'. From this and (5.15) we get,

$$tDQD'_t e = e - z_t - tDu. \tag{5.16}$$

Multiplying (5.16) by the operator $D'_t D^{-1}$, we get

$$tD'_t QD'_t e = D'_t D^{-1}(e - z_t) - tD'_t u. \tag{5.17}$$

This implies

$$\nabla f^t_{d'_t}(e) = tD'_tQD'_te + tD'_tu - e = D'_tD^{-1}(e - z_t) - e. \tag{5.18}$$

Since D^{-1} is the operator that maps $w \in E$ into $d^{-1/2}wd^{-1/2}$, we have

$$\begin{aligned} D'_tD^{-1}(e - z_t) &= D'_t(d^{-1} - d^{-1/2}z_td^{-1/2}) \\ &= d'^{1/2}_t(d^{-1} - d^{-1/2}z_td^{-1/2})d'^{1/2}_t. \end{aligned} \tag{5.19}$$

Taking trace of the equation in (5.19) and using the fact that $d'_t = d + d^{1/2}z_td^{1/2}$, and that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$, we get

$$\begin{aligned} \mathrm{Tr}(D'_tD^{-1}(e - z_t)) &= \mathrm{Tr}\big((d + d^{1/2}z_td^{1/2})(d^{-1} - d^{-1/2}z_td^{-1/2})\big) \\ &= \mathrm{Tr}(e - d^{1/2}z_td^{-1/2} + d^{1/2}z_td^{-1/2} - d^{1/2}z_t^2d^{-1/2}) \\ &= \mathrm{Tr}(e) - \mathrm{Tr}(z_t^2). \end{aligned} \tag{5.20}$$

This completes the proof of (i).

(ii) Since $\|z_t\|^2 = \sum_{i=1}^N \lambda_i^2$, where λ_i, $i = 1, \ldots, N$ are the eigenvalues of z_t, the fact that $\|z_t\| < 1$ implies $|\lambda_i| < 1$. But this implies $e + z_t \succ 0$. Hence $d'_t = D(e + z_t) \succ 0$.

(iii) It suffices to show that the matrix $B = DD'^{-1}_t\nabla f^t_{d'_t}(e)$ is negative semidefinite. Firstly, since $d'_t \succ 0$, the operator DD'^{-1}_t is well-defined. From (5.18),

$$\begin{aligned} B = e - z_t - DD'^{-1}_te &= e - z_t - d^{1/2}(d + d^{1/2}z_td^{1/2})^{-1}d^{1/2} \\ &= e - z_t - (e + z_t)^{-1}. \end{aligned} \tag{5.21}$$

It is easy to see that λ is an eigenvalue of z_t if and only if $1 - \lambda - (1 + \lambda)^{-1}$ is an eigenvalue of $e - z_t - (e + z_t)^{-1}$. Since $\|z_t\| < 1$ implies $|\lambda| < 1$, we have

$$1 - \lambda - \frac{1}{1 + \lambda} = -\frac{\lambda^2}{1 + \lambda} \leqslant 0. \tag{5.22}$$

Hence the proof of (iii).

(iv) It is easy to show that if H is a symmetric positive definite operator in $L(E, E)$ all of whose eigenvalues are bounded below by one, then for any $w \in E$,

$$\|w\|^2 \leqslant \langle w, Hw\rangle \leqslant \|Hw\|^2. \tag{5.23}$$

Now to prove the last two inequalities in (5.13) we take

$$H = \nabla^2 f^t_d(e) = t\nabla^2\phi_d(e) + I = tDQD + I, \qquad w = z_t. \tag{5.24}$$

Since $\|\nabla f^t_d(e)\| < 1$, the last two inequalities of (5.13) imply $\|z_t\| < 1$. Thus, from part (ii) we have, $d'_t = D(e + z_t) \succ 0$. This implies that $y_t(d'_t)$ and hence z'_t are well-defined. To prove the first two inequalities in (5.13), we let $H = \nabla^2 f^t_{d'_t}(e)$. To prove the third inequality in (5.13), note that if $-A \in S^n_+$, then $\mathrm{Tr}(A^2) \leqslant \mathrm{Tr}(A)^2$. Setting $A = \nabla f^t_{d'_t}(e)$, together with part (iii) of this Lemma, we get $\|\nabla f^t_{d'_t}(e)\| \leqslant \|z_t\|^2$. $\square$

Corollary 5.4. *All the results of Lemma* 5.3 *also applies to* $\psi(x)$.

Proof. When $t=1$ and $u=0$, $f^t(x)=t\phi(x)+tu^{\mathrm{T}}x+F(x)$ reduces to $\psi(x)$. $\square$

Lemma 5.5. *Fix* $\gamma_0\in(0,0.5)$. *Given* $t\in(0,\infty)$, *suppose* $d\succ 0$ *satisfies* $\|\nabla f_d^t(e)\|\leqslant\gamma_0$. *Then,* $d_t'\succ 0$, *and if* $t'=tr_*$, *where* $r_*=\left(\frac{\sqrt{N}-\gamma_0}{\sqrt{N}-\gamma_0^2}\right)$, *then* $\|\nabla f_{d_t'}^{t'}(e)\|\leqslant\gamma_0$.

Proof. That $d_t'\succ 0$ was proved in Lemma 5.3, part (ii). From Lemma 5.3, part (iv), it follows that $\|\nabla f_{d_t'}^t(e)\|\leqslant\gamma_0^2$. Let $a=D_t'QD_t'e+D_t'u$. Thus, for any $\tau\in(0,t]$, $\tau a-e=\nabla f_{d_t'}^{\tau}(e)$. We have

$$\begin{aligned}\|\nabla f_{d_t'}^{\tau}(e)\|=\left\|\frac{\tau}{t}ta-\frac{\tau}{t}e+\frac{\tau}{t}e-e\right\|&\leqslant\frac{\tau}{t}\|ta-e\|+\left(1-\frac{\tau}{t}\right)\|e\|\\&\leqslant\sqrt{N}-\frac{\tau}{t}\left(\sqrt{N}-\gamma_0^2\right).\end{aligned}$$

Next we set the right-hand-side equal to γ_0, and solve for $\tau\equiv t'$. $\square$

6. The path-following algorithm

We now have all the necessary ingredients to describe our path-following algorithm. It is based on the approximate minimization of $f^t(x)$ while decreasing t from one to zero. At $t=1$ the auxiliary vector $u=e-Qe$ turns the point $d=e$ into the minimizer of $f^1(x)$. Now given a positive point d and a value of $t\in(0,1]$ satisfying $\|\nabla f_d^t(e)\|\leqslant\gamma_0$, with $\gamma_0\in(0,0.5)$, the Newton iterate, d', obtained from one iteration as applied to the minimization of $f^t(x)$ at d will satisfy the inequality $\|\nabla f_{d'}^{t'}(e)\|\leqslant\gamma_0$, where t' is the value of t reduced by a factor r_* (see Corollary 5.4). Now Lemma 5.2 can be used to decide how to make use of the approximate minimizer of $f^t(x)$ and to what value $t_*\in(0,1)$ should we decrease t in order to obtain approximate solution to the Homogeneous Programming or to the Matrix Scaling problem. The path-following algorithm consists of two phases. The first one minimizes $f^t(x)$ while decreasing the value t. The second phase is needed only for the Matrix Scaling problem. It begins with a rough approximate solution of the Matrix Scaling, i.e. a positive d such that $\|DQDe-e\|<\gamma_0$ and continues to minimize $\psi(x)$ until an ϵ-approximate solution of the Matrix Scaling is obtained.

The path-following algorithm takes an input t_*. If ϵ-approximate Homogeneous Programming is the problem of interest, t_* will be selected so that it satisfies $C(\gamma_0)t_*=\epsilon$, where $C(\gamma_0)=\frac{1}{\gamma_0^2}[2N+\gamma_0(\sqrt{N}+1)]\|u\|^2$ (Lemma 5.2). If ϵ-approximate Matrix Scaling is the problem of interest t_* will be selected to satisfy $C(\gamma_0)t_*=\frac{1}{2}\mu$ (any positive lower bound to μ can also be used). Formally, *Initialization.* Set $u = e - Qe$, $t = 1$, $d = e$, $r_* = \left(\frac{\sqrt{N}-\gamma_0}{\sqrt{N}-\gamma_0^2}\right)$, where γ_0 is a fixed number in $(0, 0.5)$. Input $t_* \in (0, 1)$. If ϵ-approximate Matrix Scaling problem is the problem of interest input $\epsilon \in (0, 1)$.

Phase I. While $t > t_*$, replace (d, t) with (d'_t, t'), $d'_t = d + y_t(d)$, $t' = r_* t$ and repeat $(y_t(d) = -\nabla^2 f^t(d)^{-1} \nabla f^t(d))$.

Phase II. Let $\hat{d} = \sqrt{t}d$. While $\|\widehat{D}Q\widehat{D}e - e\| > \epsilon$, replace $\hat{d}$ with the Newton iterate $\hat{d}' = \hat{d} + y(\hat{d})$, with respect to minimization of $\psi(x)$ and repeat $(y(\hat{d}) = -\nabla^2\psi(\hat{d})^{-1}\nabla\psi(\hat{d}))$.

Theorem 6.1. *If $\mu = 0$, the algorithm solves ϵ-approximate Homogeneous Programming in* $\mathrm{O}\left(\sqrt{N}\ln\frac{N\|u\|}{\epsilon}\right)$ *iterations of Phase I. If $\mu > 0$, the algorithm solves ϵ-approximate Matrix Scaling in* $\mathrm{O}\left(\sqrt{N}\ln\frac{N\|u\|}{\mu} + \ln\ln\frac{1}{\epsilon}\right)$ *combined iterations of Phase I and Phase II.*

Proof. For a given $t_* \in (0, 1)$, let the kth iterate of Phase I be denoted by (d^k, t_k). Since $(d^0, t_0) = (e, 1)$, $\|\nabla f^{t_0}_{d^0}(e)\| = 0$. Thus, Lemma 5.5 implies that for all k, $d^k \succ 0$, $\|\nabla f^{t_k}_{d^k}(e)\| \leqslant \gamma_0$, and $t_k = r_*^k \leqslant \exp(k(r_* - 1))$. Thus, if the number of iterations of Phase I is k_1, then $t_* \leqslant \exp(k_1(r_* - 1))$. This implies $k_1 = \mathrm{O}\left(\sqrt{N}\ln\frac{1}{t_*}\right)$.

If $\mu = 0$, from Lemma 5.1, and Lemma 5.2 it follows that $\phi(d^k/\|d^k\|) \leqslant C(\gamma_0)t_k$. Thus, to solve ϵ-approximate Homogeneous Programming it suffices to choose t_* satisfying $C(\gamma_0)t_* = \epsilon$. From the definition of $C(\gamma_0)$, we have $\frac{1}{t_*} = \mathrm{O}\left(\frac{N\|u\|^2}{\epsilon}\right)$. The claimed complexity for ϵ-approximate Homogeneous Programming follows.

If $\mu > 0$, then for all $k > 0$ we have $\phi(d^k/\|d^k\|) \geqslant \mu$. Now from Lemma 5.2, if we implement the Path-Following algorithm with t_* satisfying $C(\gamma_0)t_* = \frac{1}{2}\mu$, then Phase I will terminate with a point (d, t) such that $d \succ 0$, $t \leqslant t_*$, and if $\widehat{D} = \sqrt{t}D$, then $\|\widehat{D}Q\widehat{D}e - e\| \leqslant \gamma_0$. In this case $\frac{1}{t_*} = \mathrm{O}\left(\frac{N\|u\|^2}{\mu}\right)$. Hence, $k_1 = \mathrm{O}\left(\sqrt{N}\ln\frac{N\|u\|}{\mu}\right)$. Instead of using μ to determine t_* we can use any positive lower bound to it.

Denote the iterates of Phase II by $\{\hat{d}_j\}$ where $\hat{d}_0 = \hat{d}$ given above. The reader may notice that the main results proved in Section 5 do not depend on u. In other words taking $t = 1$ and redefining $u = 0$ we get $f^1(x) = \psi(x)$. Thus Corollary 5.4 applies to $\psi(x)$ as well and we have

$$\|\nabla\psi_{\hat{d}_j}(e)\| \leqslant \|\nabla\psi_{\hat{d}_{j-1}}(e)\|^2 \leqslant \cdots \leqslant \|\nabla\psi_{\hat{d}_0}(e)\|^{2^j} \leqslant \gamma_0^{2^j}. \tag{6.1}$$

Thus, the number of iterations of Phase II is $\mathrm{O}\left(\ln\ln\frac{1}{\epsilon}\right)$, determined from the inequality $\gamma_0^{2^j} \leqslant \epsilon$. $\square$

Remark 1. If $\mu > 0$, an alternative way to solve the ϵ-approximate Matrix Scaling problem is to only implement Phase I, but selecting t_* to be the solution to $C(\epsilon)t_* =$

$\frac{1}{2}\mu$. From Lemma 5.2 the algorithm will give (d, t) such that $d \succ 0$, $t \leqslant t_*$, and if $\widehat{D} = \sqrt{t}D$, then $\|\widehat{D}Q\widehat{D}e - e\| \leqslant \epsilon$. In this case $\frac{1}{t_*} = \mathrm{O}\left(\frac{N\|u\|^2}{\mu\epsilon^2}\right)$. As we see the previous approach for solving ϵ-approximate Matrix Scaling could lead to a better complexity.

The following result which can easily be verified shows that when $n = 0$, Theorem 6.1 reduces to an analogous result stated with respect to positive semidefinite symmetric matrices and the Scaling Equation reduces to the ordinary diagonal scaling. In fact the Newton iterates of the path-following algorithm for this special case reduce to the iterates of the path-following algorithm for diagonal Matrix Scaling.

Corollary 6.2. *Let Q_l be an $l \times l$ symmetric positive semidefinite matrix. Let E be the set of $l \times l$ diagonal matrices. Let $Q \in L(E, E)$ be defined according to the mapping $Qx_l \equiv \sum_{i=1}^{l} \mathrm{Tr}(\mathrm{diag}(q_i)\mathrm{diag}(x_l))\mathrm{diag}(e_i)$, where q_i is the ith row of Q_l, and e_i is the ith row of the $l \times l$ identity matrix.*

(i) *$Qx_l = 0$ for some nontrivial $x_l \geqslant 0$, if and only if $Qx = 0$, $x = \mathrm{diag}(x_l)$.*
(ii) *$D_l Q_l D_l e^l = e^l$ for some $D_l = \mathrm{diag}(d_l)$, $d_l > 0$, $e^l = (1, \ldots, 1)^{\mathrm{T}}$, if and only if $DQDe = e$, where D is the operator that maps $x \in E$ into $D_l x$.*
(iii) *$Q_l d_l > 0$ for some $d_l > 0$, if and only if $Qd \succ 0$, $d = D_l$.*

Remark 2. When solving ϵ-approximate Homogeneous Programming we do not need to know if $\mu = 0$ in advance. If we do not succeed in solving the ϵ-approximate problem via the path-following algorithm, obviously $\mu > 0$. But if ϵ-approximate Homogeneous Programming is solvable it may still be the case that $\mu > 0$. When trying to solve ϵ-approximate Matrix Scaling the knowledge of a lower bound on μ is necessary. In the next section we show how such a lower bound is related to the solution of a semidefinite programming problem. In practice one way to bypass an a priori knowledge of a lower bound is as follows. We first estimate μ to be the given ϵ and pretend to solve the ϵ-approximate Homogeneous Programming via Phase I. If we did not succeed in solving the ϵ-approximate Homogeneous Programming, then we will obtain a solution to γ_0-approximate Matrix Scaling solution. Then because of the quadratic rate of convergence, to get an ϵ-approximate Matrix Scaling solution we only need a few more Newton iterations $(\ln\ln(1/\epsilon))$, as applied to the minimization of $\psi(x)$. If however we do succeed to solve the ϵ-approximate Homogeneous Programming, then we halve ϵ and repeat this as often as necessary. This process will succeed in a finite number of halving, if μ is indeed positive. More precisely, if μ is positive and less than one, we need at most $\mathrm{O}(\ln\ln(1/\mu))$ halving steps. The process would not terminate if $\mu = 0$, but each halving will give an upper bound on μ. We close this remark by pointing out that the path-following algorithm and bisection process, trivially, leads to an algorithm for finding ϵ-approximation of the value of μ itself.

7. An algorithmic proof of matrix scaling dualities

Here we give an algorithmic proof of Scaling Duality (2.9) and Gordan's Duality (2.10). We mention that it is possible to prove this in much more generality (see [10] for techniques in proving the most general cases and [9] for special cases). The following although is not stated as a theorem of the alternative is in fact equivalent to (2.9) and (2.10):

Theorem 7.1. $\mu > 0$, *if and only if there exists* $d \succ 0$ *such that* $DQDe = e$, *if and only if there exists* $d \succ 0$ *such that* $Qd \succ 0$.

Proof. From Theorem 6.1, since $\mu > 0$, there exists a point $d \succ 0$ such that $\|DQDe - e\| \leqslant \gamma_0 < 1$. Set $d^0 = d$, and for each $k \geqslant 0$ define $d^{k+1} = d^k + y^k = d^k + D^k z^k$, where y^k is the solution to $\nabla^2 \psi(d^k) y^k = -\nabla \psi(d^k)$. Corollary 5.4 (equivalently since d^{k+1} is the Newton iterate with respect to $f^t(x) = t\phi(x) + tu^{\mathrm{T}}x + F(x)$, where $t = 1$, and $u = 0$, then part (iv) of Lemma 5.3) implies that $\|D^k QD^k e - e\|$ converges to zero. To prove that $DQDe = e$ is solvable we only need to show that the sequence $\{d^k\}$ is bounded. We have

$$\|d^{k+1}\| \leqslant \|d^k\| + \|D^k\| \, \|z^k\| \leqslant \|d^k\|(1 + \|z^k\|), \tag{7.1}$$

where the second inequality makes use of the fact that $\|D^k\| \leqslant \|d^k\|$ (see (5.9) Lemma 5.2). From Lemma 5.3, part (iv), $\|z^k\| \leqslant \|z^{k-1}\|^2$. Thus,

$$\|d^k\| \leqslant \|d^0\| \sum_{i=1}^{k} \|z^0\|^i < \frac{\gamma_0}{1-\gamma_0} \|d^0\|. \tag{7.2}$$

If $DQDe = e$ for some $d \succ 0$, then $Qd \succ 0$. The fact that the latter condition implies $\mu > 0$ was already proved in Lemma 5.1. □

8. On lower bounding μ

Here we show that if $\mu = \min\{\phi(x) = 0.5\,\mathrm{Trace}(xQx) : \|x\| = 1, x \succeq 0\}$ is positive, then we can obtain a lower bound for it in terms of the optimal value of a semidefinite programming problem.

Theorem 8.1. *Assume* $\mu > 0$. *Let*

$$\mu_1 = \min\{\mathrm{Tr}(xQx) : \|x\| = 1, x \succeq 0\},$$

$$\mu_2 = \min\{\mathrm{Tr}(xQx) : \mathrm{Tr}(x) = 1, x \succeq 0\},$$

$$\mu_3 = \min\left\{\sqrt{\mathrm{Tr}(Q)}\sqrt{\mathrm{Trace}(xQx)} : \mathrm{Tr}(x) = 1, x \succeq 0\right\},$$

$$\mu_4 = \min\{\|Qx\| : \mathrm{Tr}(x) = 1, x \succeq 0\},$$

$$\mu_5 = \min\left\{\frac{1}{\sqrt{n}}\|Qx\|_1 : \mathrm{Tr}(x) = 1, x \succeq 0\right\}.$$

Then

$$\mu_1 \geqslant \mu_2,$$

$$\mu_3 = \sqrt{\mathrm{Tr}(Q)\mu_2} \geqslant \mu_4 \geqslant \mu_5 > 0.$$

In particular, we can lower bound μ *by the optimal value of the optimization defining* μ_5 *which is a semidefinite programming problem.*

Proof. We first show that $\mu_1 \geqslant \mu_2$. For any $x \in S^n_+$ we have $\mathrm{Tr}(x) \geqslant \|x\|$. Thus for any $x \succeq 0$ such that $\mathrm{Tr}(x) = 1$, $\|x\| \leqslant 1$. Thus, $\beta = 1/\|x\| \geqslant 1$. Hence βx satisfies $\|\beta x\| = 1$, and $\mathrm{Tr}((\beta x)Q(\beta x)) = \beta^2 \mathrm{Tr}(xQx) \geqslant \mathrm{Tr}(xQx)$. Hence the proof of the first inequality. The relationship between μ_2 and μ_3 is obvious.

To prove $\mu_3 \geqslant \mu_4$, we claim the following inequality:

$$\|Qx\| \leqslant \sqrt{\mathrm{Tr}(Q)}\sqrt{\mathrm{Tr}(xQx)}, \quad \forall x \in S_n.$$

To prove our claim let u_is form a complete set of eigenvalues for Q, and let λ_is be the corresponding eigenvalues. Thus $\mathrm{Tr}(u_i u_j) = 1$ if $i = j$, and 0 if $i \neq j$. Given $x \in S_n$, $x = \sum \alpha_i u_i$. We have $Qx = \sum \alpha_i u_i$. From these it follows that

$$\|Qx\| = \sqrt{\alpha_i^2 \lambda_i^2}, \quad \mathrm{Tr}(Q) = \sum \lambda_i, \quad \mathrm{Tr}(xQx) = \sum \alpha_i^2 \lambda_i.$$

Now the following inequality is obvious

$$\sum \alpha_i^2 \lambda_i^2 \leqslant \sum \lambda_i \sum \alpha_i^2 \lambda_i.$$

Finally, to show $\mu_4 \geqslant \mu_5$, we observe that for $x \in S_n$ we have

$$\|Qx\| \geqslant \frac{1}{\sqrt{n}}\|Qx\|_1.$$

Finally, we note that μ is positive if and only if μ_5 is positive. □

9. Homogeneous programming formulation of feasibility in SDP

Consider the feasibility of the following set where here we have used uppercase letters for matrices and lowercase for vectors:

$$\begin{aligned} V = \big\{ & (X, w) : \mathrm{Tr}(C_i X) + a_i^{\mathrm{T}} w = b_i, \\ & i \in I, X \succeq 0, w \geqslant 0, (X, w) \neq (0, 0)\big\}, \end{aligned} \tag{9.1}$$

where $(X, w) \in S^n \times \Re^k$, and for $i \in I = \{1, \ldots, m\}$, $C_i \in S^n$, $a_i \in \Re^k$, $b_i \in \Re$ are given inputs.

Clearly if at least one of the b_i's is nonzero, the constraint $(X, w) \neq (0, 0)$ is redundant. In this case, if C_i's are all zero, the problem is the ordinary feasibility in linear programming, and if a_i's are all zero, it is the feasibility problem in SDP. Thus the above feasibility problem is more general than the *pure* feasibility problem in LP or SDP. If all the b_i's are zero, the problem is to find a nontrivial nonnegative point that satisfies all the equality constraints. Indeed our first goal is to convert the problem into the special case where all the b_i's are assumed to be zero. We refer to this special case as the *homogeneous feasibility problem*. When at least one of the b_is is nonzero this conversion can easily be derived, if V has no recession direction, i.e., there does not exist a nonzero (X, w) such that $X \succeq 0$, $w \geqslant 0$, and $\mathrm{Tr}(C_i X) + a_i^{\mathrm{T}} w = 0$, for all $i \in I$. In particular, if V is bounded it has no recession direction.

In this section we first show that when V has no recession direction, the feasibility of V can be formulated as a homogeneous feasibility problem. Then we show that this homogeneous feasibility problem can be converted into a Homogeneous Programming, i.e. the problem of testing if $\phi(x) = \frac{1}{2}\mathrm{Tr}(xQx)$ has a nontrivial zero $x \succeq 0$, for some symmetric positive semidefinite linear operator Q in $L(E, E)$.

Consider the problem of testing the feasibility of the set V. If all the b_i's are zero the corresponding feasibility is already in a homogeneous format. Thus assume that not all $b_i = 0$. Let

$$\overline{V} = \big\{ (X, w, \alpha) : \mathrm{Tr}(C_i X) + a_i^{\mathrm{T}} w - \alpha b_i = 0, \\ i \in I, X \succeq 0, w \geqslant 0, \alpha \geqslant 0, (X, w, \alpha) \neq 0 \big\}. \tag{9.2}$$

Let γ^* be the optimal value of the following conic linear programming:

$$\begin{aligned} &\text{minimize} \quad \gamma \\ &\text{subject to} \quad \mathrm{Tr}(C_i X) + a_i^{\mathrm{T}} w - \alpha b_i - \gamma \left(\mathrm{Tr}(C_i) + \sum_{j=1}^{k} a_{ij} - b_i \right) = 0, \\ &\qquad\qquad i = 1, \ldots, m, \\ &\qquad\qquad \mathrm{Tr}(X) + \sum_{i=1}^{k} w_i + \alpha + \gamma = n + k + 2, \\ &\qquad\qquad X \succeq 0, w \geqslant 0, \alpha \geqslant 0, \gamma \geqslant 0. \end{aligned}$$

Note that $X = e$, $w = (1, \ldots, 1)^{\mathrm{T}}$, $\alpha = \gamma = 1$ is an interior feasible point. The following trivial lemma can now be stated.

Lemma 9.1. *Assume that V has no recession direction. Then, $V \neq \emptyset$ if and only if $\overline{V} \neq \emptyset$, if and only if $\gamma^* = 0$.*

Proof. Clearly, if $X \in V$, then $(X, w, 1) \in \overline{V}$. Conversely, suppose that $(X, w, \alpha) \in \overline{V}$. If $\alpha = 0$, then (X, w) is necessarily a recession direction in V. Thus $\alpha > 0$. Clearly, $\alpha^{-1}(X, w) \in V$. To prove the second equivalence, note that an appropriate scalar multiple of any point in $\overline{V}$ is an optimal solution of the conic LP, with $\gamma^* = 0$. Conversely, if $\gamma^* = 0$, any optimal solution of the conic LP is a point in $\overline{V}$. □

For $i = 1, \dots, m$, let $B_i = \mathrm{diag}(a_{i1}, \dots, a_{ik}, -b_i)$, i.e. the $(k+1) \times (k+1)$ diagonal matrix, and let $c_i = \mathrm{diag}(C_i, B_i)$. Let E be the subspace of S^{n+l} described in (2.1), where $l = k+1$, and define

$$\widehat{V} = \{x \in E : \mathrm{Tr}(c_i x) = 0, \quad i = 1, \dots, m, \ \ x \succeq 0, \ \ x \neq 0\}, \tag{9.3}$$

$$\begin{aligned} &\phi(x) = \frac{1}{2}\mathrm{Tr}(xQx), \quad Qx = \sum_{i=1}^{m} \mathrm{Tr}(c_i x) c_i, \\ &\mu = \min\{\phi(x) : \|x\| = 1, \ \ x \succeq 0\}. \end{aligned} \tag{9.4}$$

Note that Q is a symmetric linear operator in $L(E, E)$ and positive semidefinite, i.e. for any $x \in E$, $\phi(x) \geqslant 0$. We now state the following lemma which in particular implies that the feasibility of V is reducible to a Homogeneous Programming problem.

Lemma 9.2. *Assume that V has no recession direction. $V \neq \emptyset$, if and only if $\widehat{V} \neq \emptyset$, if and only if $\mu = 0$.*

Proof. Clearly, $\overline{V} \neq \emptyset$ if and only if $\widehat{V} \neq \emptyset$. Now apply Lemma 9.1. Since $\mathrm{Tr}(xQx) = 0$, if and only if $\mathrm{Tr}(c_i x)^2 = 0$, for all i, the proof is complete. □

The problem of testing if γ^* is zero is reminiscent of Karmarkar's canonical LP problem. However its polynomial time solvability is unknown as is the case with the general SDP problem. The complexity issues of SDP in the Turing machine model and the real number model are discussed in Ramana [19]. He shows that given a SDP there exists a dual problem with polynomially many variable and polynomial size coefficients. In particular, from his results it follows that in the Blum et al. [3] real number model of computation semidefinite programming problem belongs to NP and Co-NP. It should be noted that our dualities imply that the problem of testing the solvability of Scaling Equation for a symmetric positive semidefinite linear operator is also both NP and Co-NP.

Most algorithms for SDP, as well the one exhibited in this paper are concerned with approximate solutions to the underlying SDP where the number of iterations is polynomial in the dimension n, $\ln(1/\epsilon)$, and the size of encoding of constants that depend on the input data. For instance to solve the ϵ-approximate version of the problem we test if $\gamma_* \leqslant \epsilon$. This approximate problem can be solved via various path-following algorithms, e.g. using Nesterov and Nemirovskii's theory of self-concordance [15]. Also by modification of any existing algorithm for SDP, e.g. those described in Alizadeh [1], Alizadeh et al. [2], Nesterov and Todd [16], and

Vandenberghe and Boyd [20]. It can also be solved by the path-following algorithm described by Kalantari [10] for self-concordant Homogeneous Programming. The general Homogeneous Programming is the problem of testing if a homogeneous function has a nontrivial zero over a closed convex pointed cone and its intersection with a subspace of the underlying space. The latter algorithm requires that the homogeneous degree of the objective function is strictly larger than one. But for the conic LP above, all is needed is to replace the linear objective function by its square. All the above path-following algorithms exhibit the same theoretical iteration complexity in obtaining an ϵ-approximate solution.

In view of unknown complexity results in SDP, all the above algorithms, including the path-following algorithm of this paper, face the same difficulty in deciding on the exact value of γ^*, since unlike linear programming and diagonal Matrix Scaling, the polynomial-time solvability of the problem, in the exact sense, is an open question. One of the goals of the present paper was to show that the problem of testing if V is nonempty can be formulated as a canonical quadratic Homogeneous Programming problem which in turn can be solved via a path-following algorithm that at the same time is capable of solving the Matrix Scaling problem over the cone of positive definite symmetric matrices.

In fact, at least in theoretical sense, and just as in the case of LP [11], the algorithm of this paper can be used to solve the general SDP problem. Firstly, if the feasibility problem in SDP is not known to have a nonempty recession one can impose a bound on the feasible region. Theoretical bounds, possibly doubly exponential have been shown to exist on the norm of solutions [18]. Indeed just as in the case of linear programming, any algorithm for the homogeneous feasibility problem can be turned into an algorithm for the general SDP itself. This can be done in the same fashion as for LP (see e.g. [11] where it is shown how to solve LP, given any feasibility algorithm, with or without the use of conic LP duality).

10. Concluding remarks

In this paper we described a simple path-following algorithm that either finds an ϵ-approximate nontrivial nonnegative zero of a positive semidefinite symmetric linear operator Q, or finds an ϵ-approximate solution to the Scaling Equation $DQDe = e$. Moreover we proved dualities that relate Homogeneous Programming to Matrix Scaling. In particular, our algorithm can be used to solve a canonical feasibility problem in SDP (and even SDP itself). Hence our dualities also give new dualities for this canonical feasibility in SDP. Our dualities in particular imply that in the real model number of computation, to test the solvability of Matrix Scaling is both in NP and Co-NP This follows because to test if $Qd \succ 0$ can be done efficiently in such model of computation. The Matrix Scaling results and a corresponding algorithm can easily be extended to prove that in the absence of the solvability of the Homogeneous Programming, the Scaling Equation $DQDe = \lambda$ is solvable for any $\lambda \in S_+^n$. One of our dualities suggests that within each iteration of the algorithm we can test if $Qd \succ 0$. If so, it follows that $\mu > 0$. In the context of the feasibility problem in SDP this is a useful test for infeasibility. Potentially, the algorithm of this paper can be stated with the same degree of simplicity for analogous problems where S_+^n may be replaced with the second-order cone (Lorentz cone), or even with an arbitrary symmetric cone. For the definition and properties of symmetric cones, see Faraut and Koranyi [5], Güler [6], and Nesterov and Todd [16]. For more general definition of scaling and its relationship to convex programming, see [10]. The path-following algorithm described in the latter paper for general self-concordant Homogeneous Programming also shows how to approximate the minimum value of the logarithmic potential function $\psi(x)$, as well as the homogeneous (Karmarkar) potential function $\phi(x)/(\det(x))^{2/n}$ to a prescribed accuracy ϵ.

A referee has remarked that the feasibility problem in SDP has been treated in the literature with techniques that does not need to make any assumptions on the recession cone, e.g. in [21, Chapter 5] and [4]. In contrast, our algorithm, when viewed as an algorithm for solving the feasibility problem in semidefinite programming requires the recession direction to be empty, a restriction which allows the conversion of the problem into a Homogeneous Programming problem. We have a few comments with this regard. The main goal of this paper is the proof of the duality and the complexity results pertaining to the solvability of ϵ-approximate solution to the two problems defined here: Homogeneous Programming and the Matrix Scaling problems. Our algorithm is capable of handling two tasks at the same time: testing ϵ-approximate Homogeneous Programming or Matrix Scaling. This is because the Matrix Scaling problem is a genuine *dual* to Homogeneous Programming. The algorithm makes vast use of these dualities. Moreover, the main results needed to prove correctness of the path-following algorithm is based on the applications of elementary linear algebra on positive (semi)definite symmetric matrices. This is a desirable feature of the algorithm since when viewed as a feasibility algorithm for homogeneous SDP, it has the same theoretical complexity as those of the best-known algorithms for general SDP. Furthermore, our path-following algorithm is capable of solving the Matrix Scaling problem, a problem the existing SDP algorithms are not designed to solve. The practicality of the algorithm is neither the claim nor the concern of the present paper.

Each of the two problems, the Homogeneous Programming and the Matrix Scaling problems are of course convex programming problems and hence can be handled by any convex programming algorithm. However, while Homogeneous Feasibility is equivalent to an SDP, it is not clear if the Matrix Scaling problem can be formulated as an SDP. It has been observed that the ordinary diagonal matrix scaling problem for positive definite matrices can be formulated as a second-order cone programming. But that sort of formulation does not make the problem any easier, neither with respect to its complexity in the Turing machine model of computation, nor from the practical point of view. Regardless of the formulations, scaling dualities must be employed.

Just as the diagonal matrix scaling problem does not appear to benefit from formulation as a second-order cone program, the Matrix Scaling problem considered in this paper, in the best case, can be formulated as a self-concordant convex programming over a pointed closed convex cone. However, it is not clear if any such formulation would have any additional benefits. Not only any algorithm for whichever formulation of the problem must (implicitly or explicitly) make use of the Scaling Dualities stated here, but as shown in [10] the Homogeneous Programming and the Matrix Scaling problems can be defined in much more general setting. This general setting in particular includes the case of general self-concordant Homogeneous Programming itself. This allows the validity of an analogous path-following algorithm whose correctness and analysis of complexity requires much more sophisticated tools than those stated in this paper. We close by remarking that since most convex programming problems can be converted to a general Homogeneous Programming and since a corresponding Matrix Scaling problem can be stated, the Matrix Scaling problem remains to be an important relative of convex programming worthy of further research, algorithmically or otherwise.

Acknowledgments

We are grateful to an anonymous referee for raising the question of lower bounding the quantity μ which motivated the results of Section 8 and for other very useful remarks that have improved the readability and correctness of the paper.

References

[1] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, SIAM J. Optim. 5 (1995) 13–51.
[2] F. Alizadeh, J.-P. Haeberly, M. Overton, Primal–dual interior point methods for semidefinite programming, SIAM J. Optim. 8 (1998) 746–768.
[3] L. Blum, M. Shub, S. Smale, On a theory of computational and complexity over the real numbers: NP-Completeness, recursive functions and universal machines, Bull. AMS 21 (1989) 1–46.
[4] E. de Klerk, C. Roos, T. Terlaky, Initialization in semidefinite programming via a self-dual skew-symmetric embedding, Operat. Res. Lett. 20 (1997) 213–221.
[5] J. Faraut, A. Koranyi, Analysis on Symmetric Cones, Oxford University Press, New York, 1994.
[6] O. Güler, Barrier functions in interior point methods, Math. Operat. Res. 21 (1996) 860–885.
[7] B. Kalantari, Canonical problems for quadratic programming and projective methods for their solution, Proceedings of AMS Conference "Mathematical Problems Arising from Linear Programming", 1988, in: J.C. Lagarias, M. Todd (Eds.), Contemporary Mathematics, vol. 114, 1990, pp. 243–263.
[8] B. Kalantari, Derivation of a generalized and strengthened Gordan theorem from generalized Karmarkar potential and logarithmic barrier functions, Technical Report LCSR-TR-121, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1989.
[9] B. Kalantari, A theorem of the alternative for multihomogeneous functions and its relationship to diagonal scaling of matrices, Linear Algebra Appl. 236 (1996) 1–24.

[10] B. Kalantari, Scaling dualities and self-concordant homogeneous programming in finite dimensional spaces, Technical Report LCSR-TR-359, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1999.
[11] B. Kalantari, M.R. Emamy-K, On linear programming and matrix scaling over the algebraic numbers, Linear Algebra Appl. 262 (1997) 283–306.
[12] N. Karmarkar, A new polynomial time algorithm for linear programming, Combinatorica 4 (1984) 373–395.
[13] L. Khachiyan, B. Kalantari, Diagonal matrix scaling and linear programming, SIAM J. Optim. 4 (1992) 668–672.
[14] A.W. Marshall, I. Olkin, Scaling of matrices to achieve specified row and column sums, Numer. Math. 12 (1968) 83–90.
[15] Y. Nesterov, A.S. Nemirovskii, Interior-Point Polynomial Algorithms in Convex Programming, SIAM, Philadelphia, PA, 1994.
[16] Y.E. Nesterov, M.J. Todd, Self-scaled barriers and interior-point methods for convex programming, Math. Operat. Res. 22 (1997) 1–42.
[17] D.P. O'Leary, Scaling symmetric positive definite matrices to prescribed row sums, Linear Algebra Appl. 370 (2003) 185–191.
[18] L. Porkolab, L. Khachiyan, On the complexity of semidefinite programs, J. Global Optim. 10 (1997) 351–365.
[19] M.V. Ramana, An exact duality theory for semidefinite programming and its complexity implications, Math. Programming 77 (Ser. B) (1997) 129–162.
[20] L. Vandenberghe, S. Boyd, Semidefinite programming, SIAM Rev. 38 (1996) 49–95.
[21] H. Wolkowicz, R. Saigal, L. Vandenberghe, Handbook of Semidefinite Programming: Theory, Algorithms, and Applications, Kluwer Academic Publishers, Boston, MA, 2000.